北京量子信息科学研究院
Beijing Academy of Quantum Information Sciences

量子计算在核物理中的应用研究

魏世杰

北京量子信息科学研究院

2024年8月6日

目录

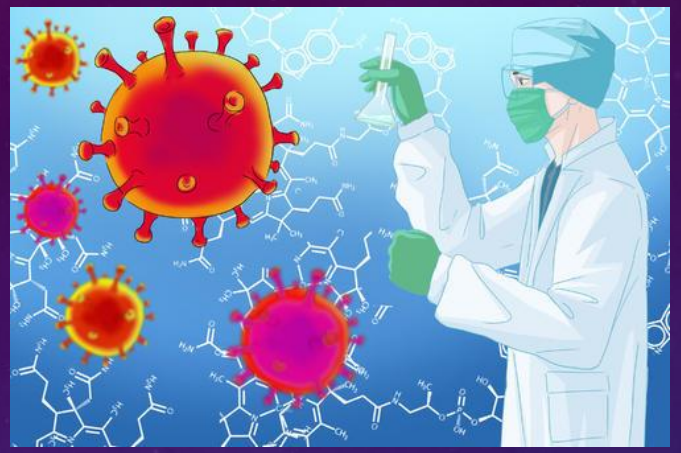
- 量子计算背景与进展
- 量子算法框架
- LCU算法模式
- 量子模拟—原子分子基态能量计算

一. 量子计算背景与进展

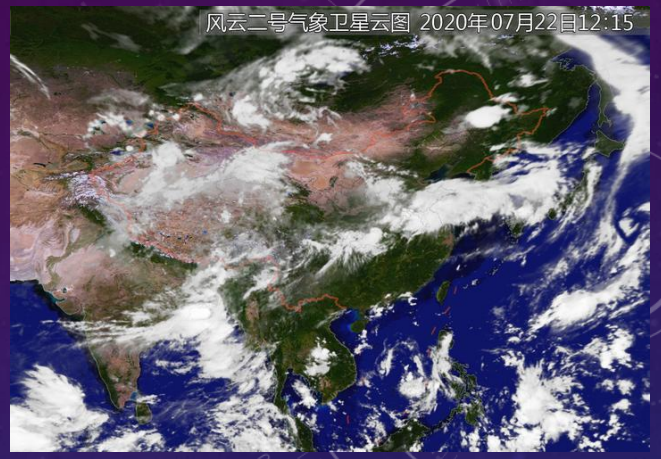
北京量子信息科学研究院
Beijing Academy of Quantum Information Sciences

算力非常重要，关系到国计民生，国家安全。用算盘计算原子弹。

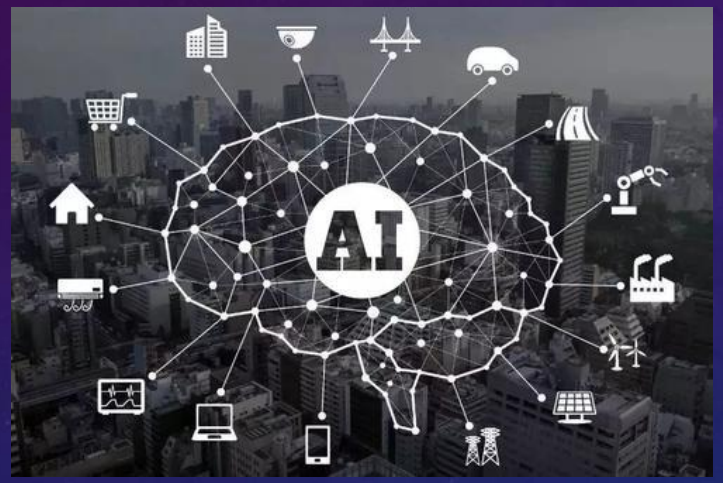
密码分析



药物研发



气象预报



人工智能

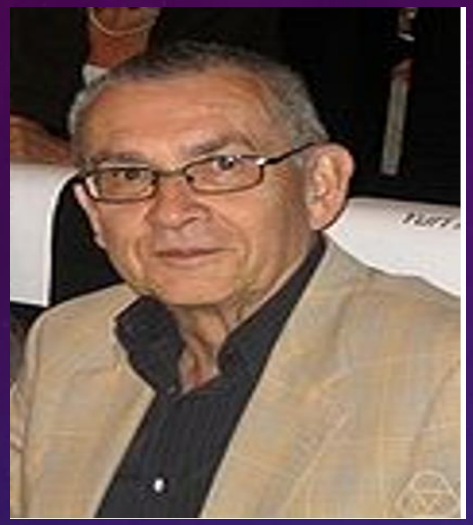
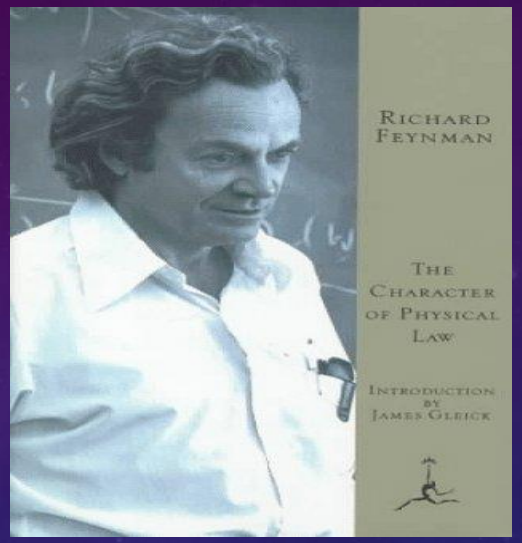
战场分析

银行金融

量子计算机可大幅度提升算力，用于物质研究、密码分析、材料设计、药物研发、人工智能、气象预报、战场分析、指挥决策、大数据分析、银行金融…

北京量子信息科学研究院
Beijing Academy of Quantum Information Sciences

提出和建立量子计算概念

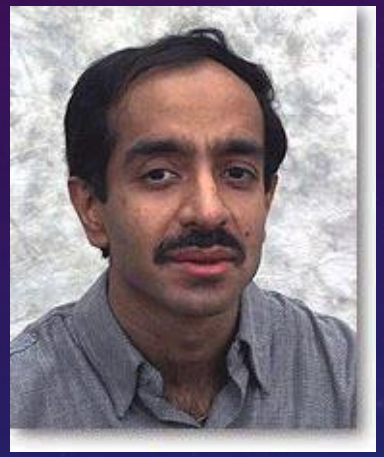




- 1980年，Benioff （美国，1930.5.1）Manin（德国，1937.2.12；）提出量子计算概念。
- 1981年，Feynman（美国，1918.5.11-1988.2.15； 诺贝尔奖）提出量子模拟。
- 1985年，Deutsch(英国，1953.5.18)证明量子计算普适性。

北京量子信息科学研究院
Beijing Academy of Quantum Information Sciences

量子算法突破，引发大规模国际研究

Shor Algorithm（94），指数加快大数分解：300 位的大数，如果每秒运算10^{12} 次， 经典计算机需要 15万 年; 量子计算机只需要 1秒.

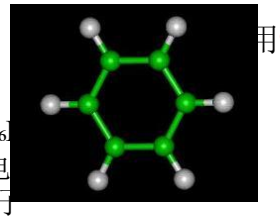


Grover /Long Algorithm（96/01），平方根加速无序数据库搜索：搜索 10^{24} 样本的数据库，如果每秒搜索10^{12} 次，经典计算机需要 2万年，量子计算机只需要 11 小时。

量子模拟

量子计算机

用96比特超导量子计算机只需14小时



苯环 (C_6
72个可分辨电
42个电子

经典计算机

神威太湖超级计算机需要10^{18}年（100亿亿年）

量子模拟（n体系的大小）

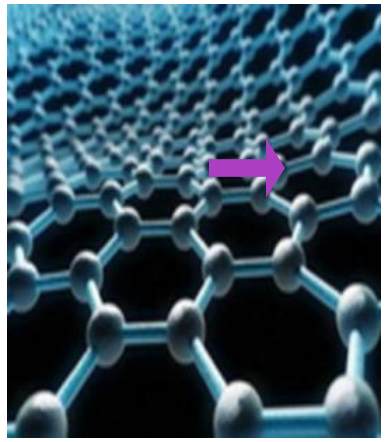
经典计算机：$\approx 2^{2n}$ 次步骤

量子计算机：$\approx n^4$ 次步骤

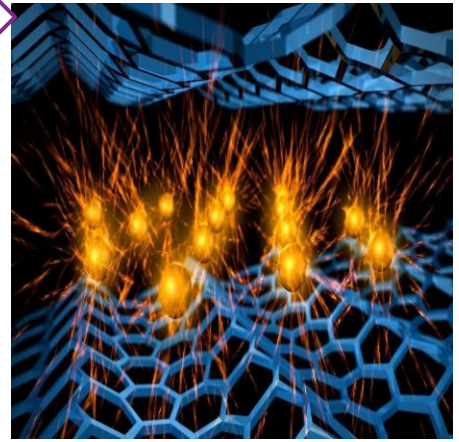
对于苯环 n=72

量子模拟的用途

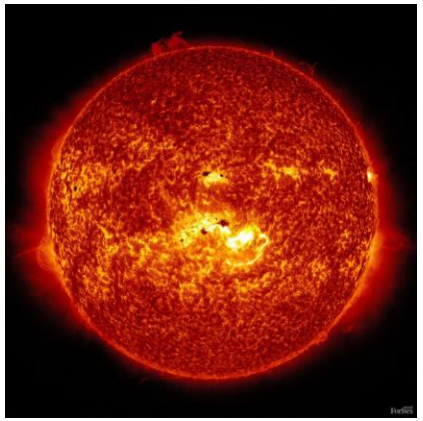
新药物研发



新材料设计



高温超导机理探究



实用核聚变

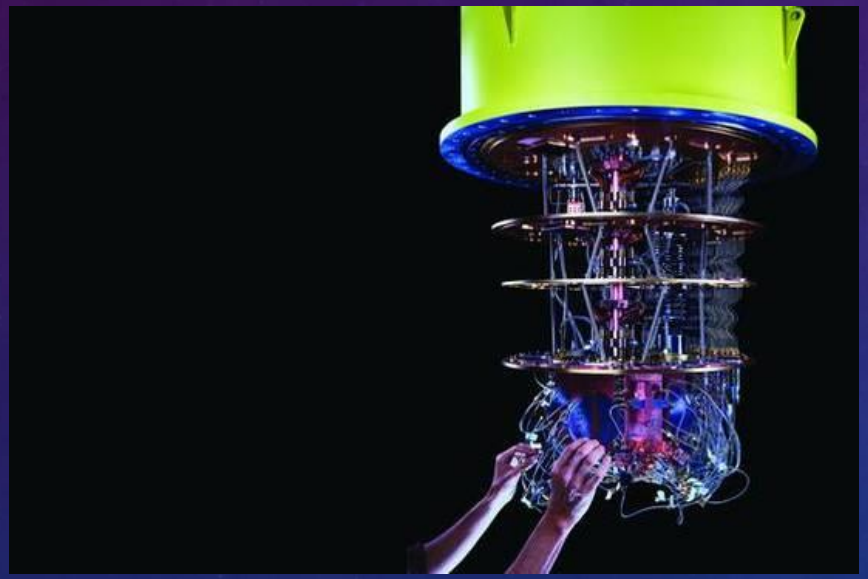
谷歌的量子霸权实验：NISQ时代全面开启

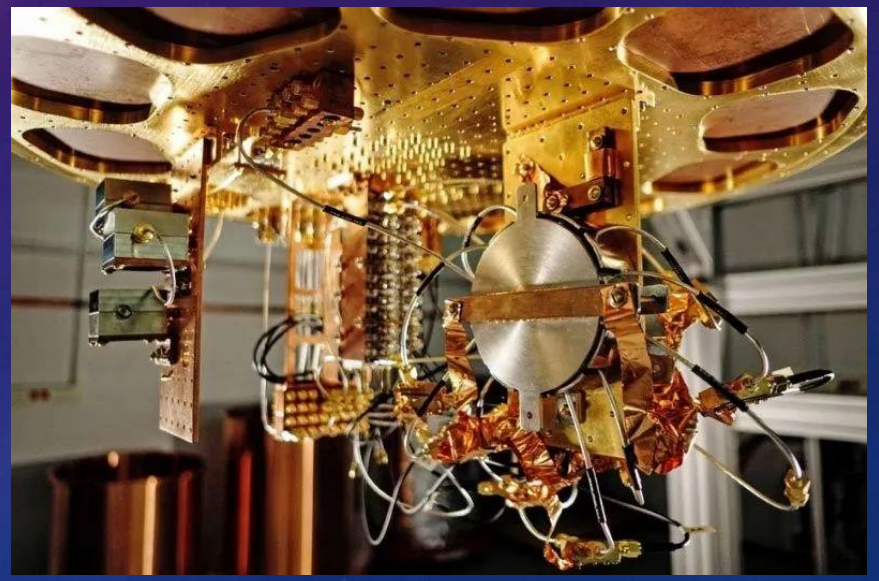
运算任务：对一个量子随机数生成器的输出进行采样

量子计算机（谷歌**Sycamore**，**53**量子比特）：**200**秒

传统计算机（**Summit**超算，**100**万**CPU**）：**~1**万年

2019年
10月24日





Google AI Quantum et al., “Quantum supremacy using a programmable superconducting processor,” Nature 574, 505 (2019).

有噪中规量子计算时代 （NISQ）

利用量子计算机高效地解决特定的科学问题即将成为现实，并且成为今后持续的研究热点和应用方向。应用层面实现量子优势的具体问题：

- 可应用于材料合成与生物制药的量子化学模拟；
- 可应用于机器学习与组合优化的量子优化算法。

- 应用瓶颈： 现阶段量子计算硬件比特个数少，操作精度低，噪声影响大，导致无法输出正确计算结果。

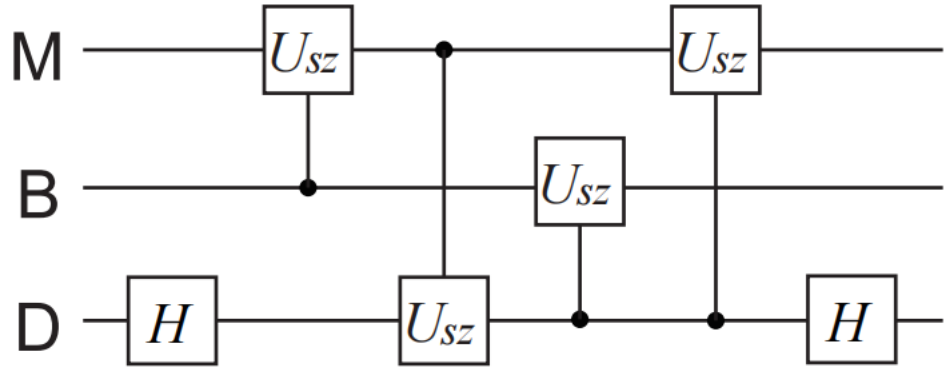
二. 量子算法框架

量子计算机的基本架构

量子比特	→	初始态的制备	→	量子操控	→	量子测量
二能级的量子体系自旋、自然原子、人工体系		所有比特的基态、叠加态、多比特的纠缠态（GHZ态、W态）		单比特量子门，两比特纠缠门，多比特控制门		测量量子比特读出计算结果（确定性，非破坏性）

量子计算机的计算模型

- Circuit model



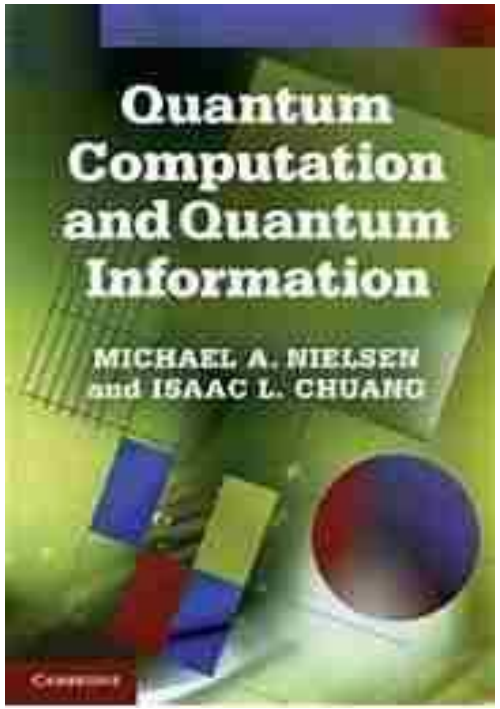
- Adiabatic quantum computation (Adiabatic theorem)

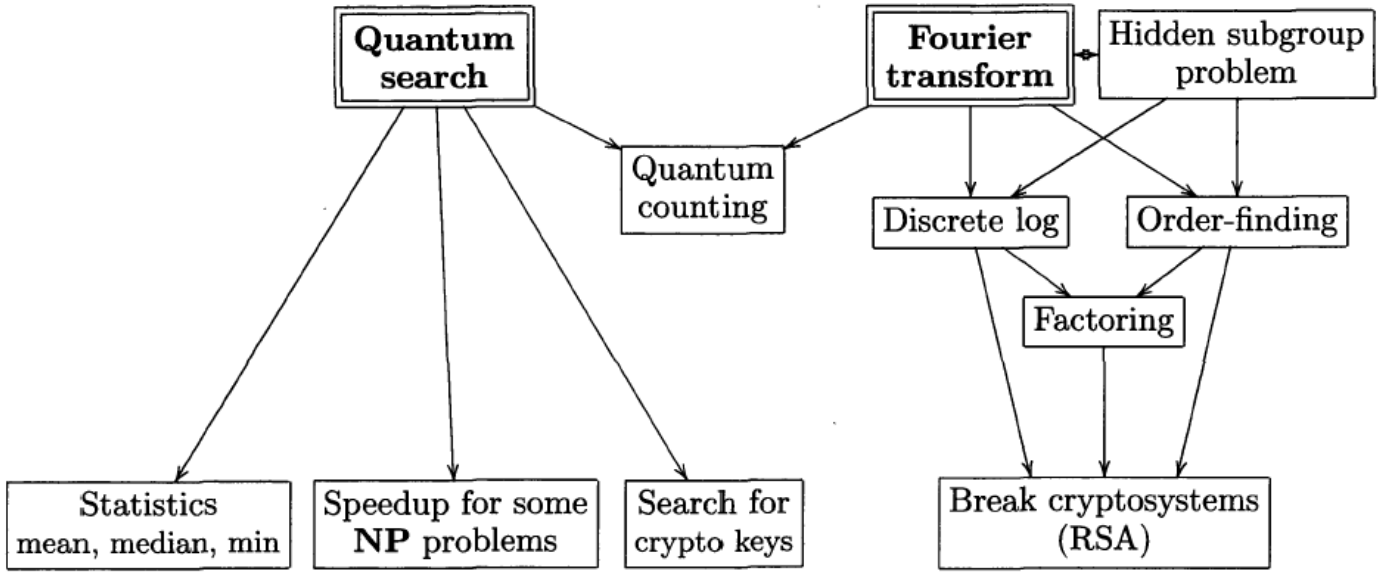
$$H(t) = (1-t)H_i + tH_f$$

$|\psi(0)\rangle$ is an eigenstate of H_i; $|\psi(1)\rangle$ is an eigenstate of H_f.

- Measurement-based (one-way) quantum computation (cluster states)

主要的量子算法

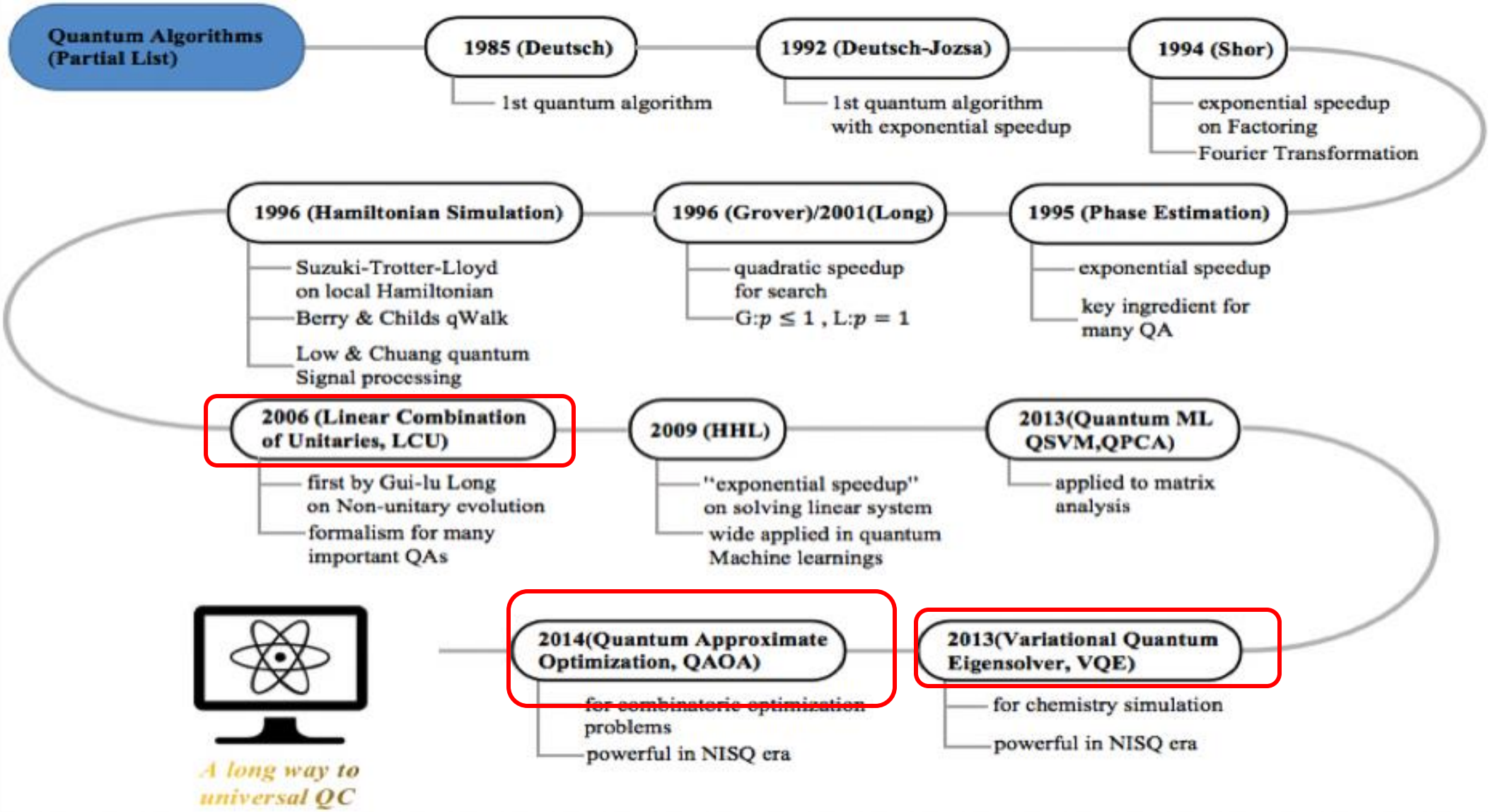




Grover算法

Shor算法

◆ 量子算法整体发展脉络


Quantum Algorithms (Partial List)
1985 (Deutsch)
1st quantum algorithm
1992 (Deutsch-Jozsa)
1st quantum algorithm with exponential speedup
1994 (Shor)
exponential speedup on Factoring
Fourier Transformation
1995 (Phase Estimation)
exponential speedup
key ingredient for many QA
1996 (Grover)/2001(Long)
quadratic speedup for search
G:$p \leq 1$, L:$p = 1$
1996 (Hamiltonian Simulation)
Suzuki-Trotter-Lloyd on local Hamiltonian
Berry & Childs qWalk
Low & Chuang quantum Signal processing
2006 (Linear Combination of Unitaries, LCU)
first by Gui-lu Long on Non-unitary evolution
formalism for many important QAs
2009 (HHL)
"exponential speedup" on solving linear system
wide applied in quantum Machine learnings
2013(Quantum ML QSVM,QPCA)
applied to matrix analysis
2013(Variational Quantum Eigensolver, VQE)
for chemistry simulation
powerful in NISQ era
2014(Quantum Approximate Optimization, QAOA)
for combinatoric optimization problems
powerful in NISQ era
A long way to universal QC

三. LCU算法模式

传统量子算法

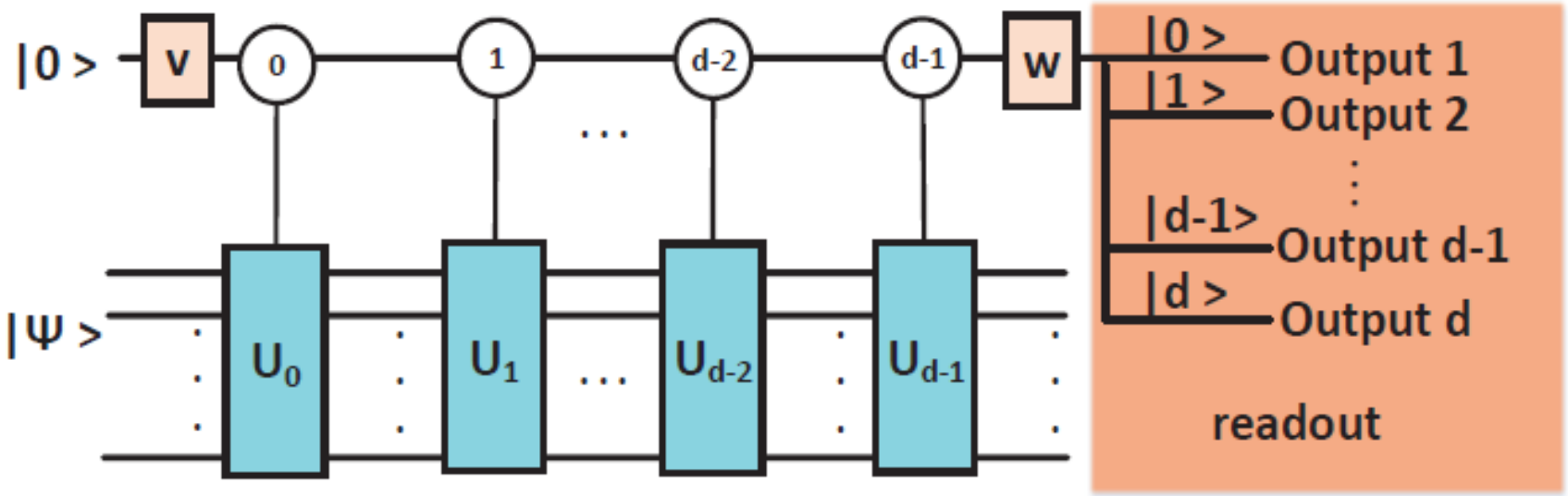
- 传统的量子算法都是使用酉算符乘积的形式进行信息的处理。
- 传统的算法都是通过幺正演化实现。
- 量子初态经过一系列先后的酉操作得到末态：

$$|\psi_f\rangle = U_n U_{n-1} \dots U_1 |\psi_i\rangle$$

对偶量子算法

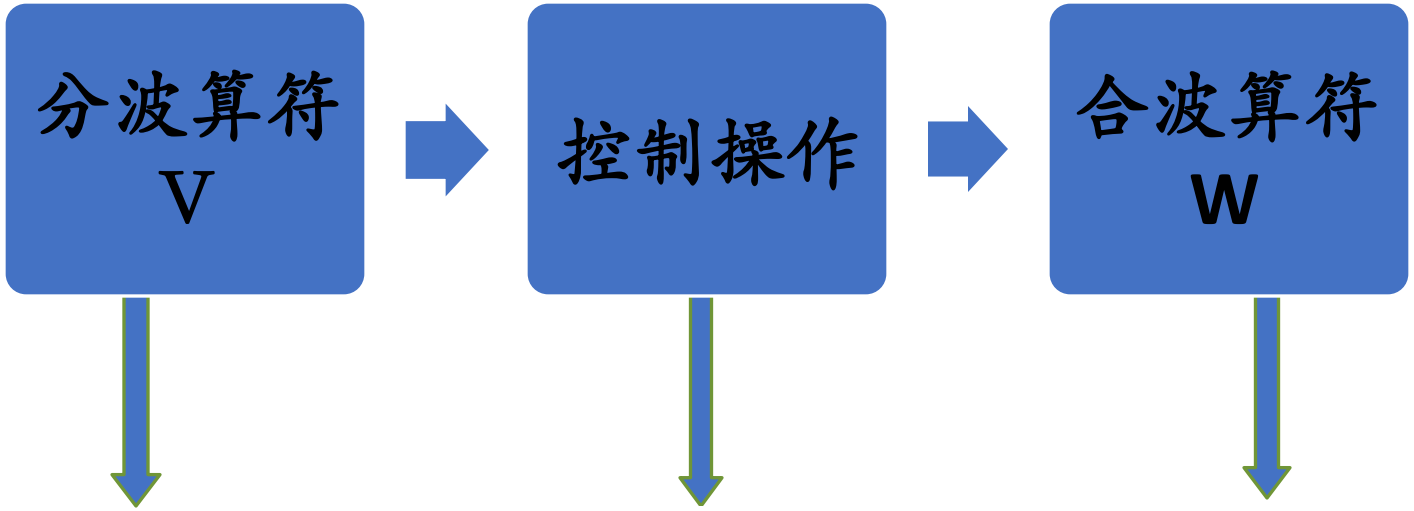
- 对偶量子算法可以使用酉算符线性叠加（LCU）的形式进行信息的处理。
- 量子初态经过一系列平行的的酉操作叠加得到末态
- 对偶量子计算实现了酉算子的加减乘除运算，可以连接经典算法和量子算法

$$|\psi_f\rangle = \left(\sum_j \alpha_j U_j\right)|\psi_i\rangle$$



Gui-Lu L. General quantum interference principle and duality computer[J]. Communications in Theoretical Physics, 2006

基本概念



$$D_m|\psi\rangle = \oplus_{i=0}^{d-1}(p_i|\psi\rangle_i)\,U_i \otimes |i\rangle\langle i|$$

$$C_m(|\psi_0\rangle_0 \oplus \cdots \oplus |\psi_{d-1}\rangle_{d-1}) = \sum_{i=0}^{d-1} q_i|\psi_i\rangle_0$$

对偶量子门

成功概率

$$L_c = \sum_{i=0}^{d-1} c_i U_i$$

$$P_s = \langle\Psi| \left(\sum_{i_1} W_{0i_1} V_{i_1 0} U_{i_1}\right)^{\dagger} \left(\sum_{i_2} W_{0i_2} V_{i_2 0} U_{i_2}\right) |\Psi\rangle$$

简单实例

1. 初始化 $|\varphi\rangle|0\rangle$

2. 分波操作: Hadamard 门 $|\varphi\rangle\frac{|0\rangle+|1\rangle}{\sqrt{2}}$

3. 作用0和 1 控制的算符 U_0 和U_1 $\frac{U_0|\varphi\rangle|0\rangle+U_1|\varphi\rangle|1\rangle}{\sqrt{2}}$

4. 合波操作:再次作用Hadamard 门 $\frac{U_0|\varphi\rangle+U_1|\varphi\rangle}{2}|0\rangle+\frac{U_0|\varphi\rangle-U_1|\varphi\rangle}{2}|1\rangle$

5. 放大|0> 态的振幅, 或者直接对辅助比特做测量，如若测量为0，测量工作比特，否则重新进行算法过程。

四.量子模拟—原子分子基态能量计算

原子核壳模型的全量子计算包

研究背景：复杂原子分子无法用经典计算机精确模拟和计算(指数难问题!)

解决方案：提出了基于全量子本征求解器(FQE)的原子核壳模型的全量子计算包（QCSH），可在量子计算机上高效进行原子分子模拟。

$$H = \sum_{i=1}^{Z} \frac{\vec{p}_i^2}{2M_p} + \sum_{j=Z+1}^{A} \frac{\vec{p}_j^2}{2M_n} + \sum_{k=1}^{A} V(\vec{r}_k) + \sum_{1 \le i_1 < i_2 \le Z} \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\vec{r}_{i_1} - \vec{r}_{i_2}|} + \frac{1}{2} \sum_{k_1 \neq k_2} v(k_1, k_2).$$

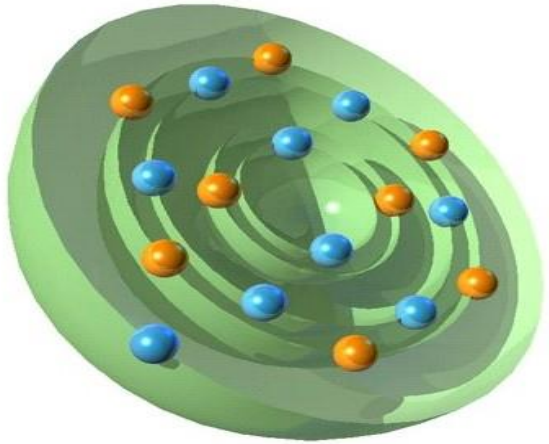
Second quantization

$$H = \sum_{ij} h_{ij} a_i^\dagger a_j + \frac{1}{2} \sum_{ijkl} h_{ijkl} a_i^\dagger a_j^\dagger a_k a_l$$

J-W transformation　　B-K transformation

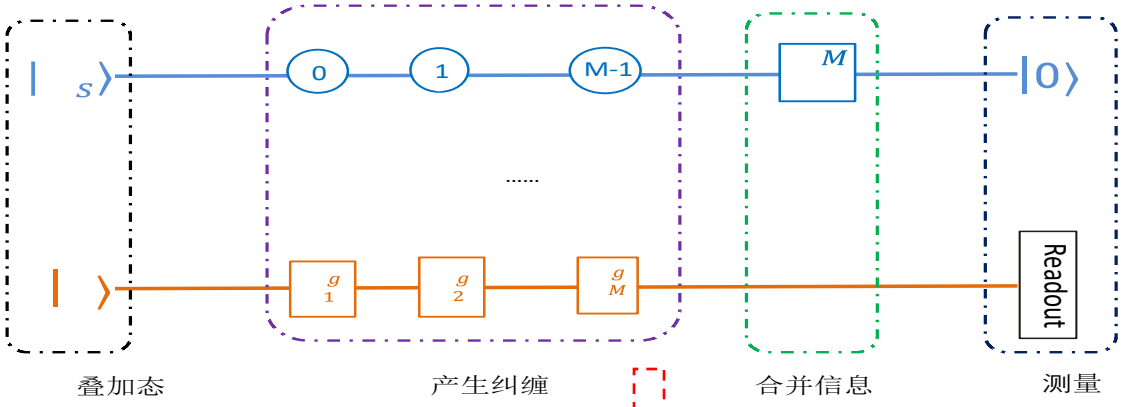
$$H = \sum_{i,\alpha} h_\alpha^i \sigma_\alpha^i + \sum_{i,j,\alpha,\beta} h_{\alpha\beta}^{ij} \sigma_\alpha^i \sigma_\beta^j + \dots$$

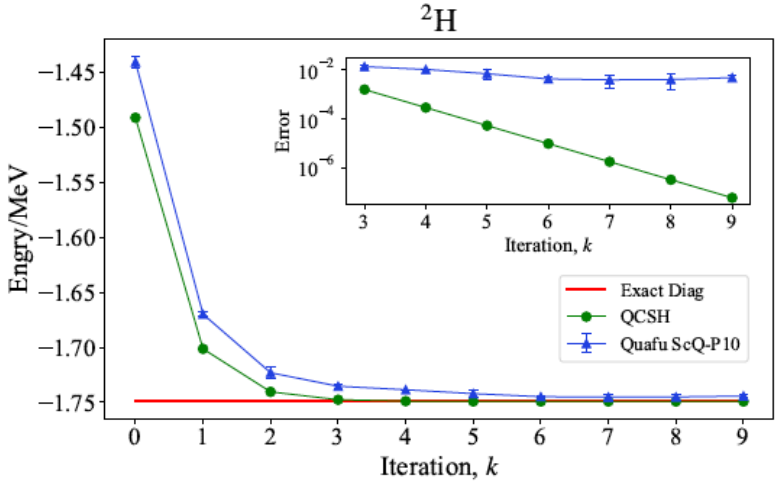
利用幺正算符线性叠加（LCU）实现

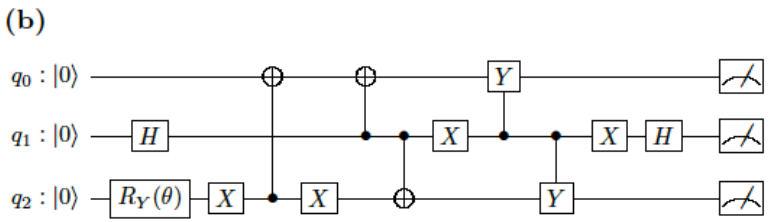


原子核

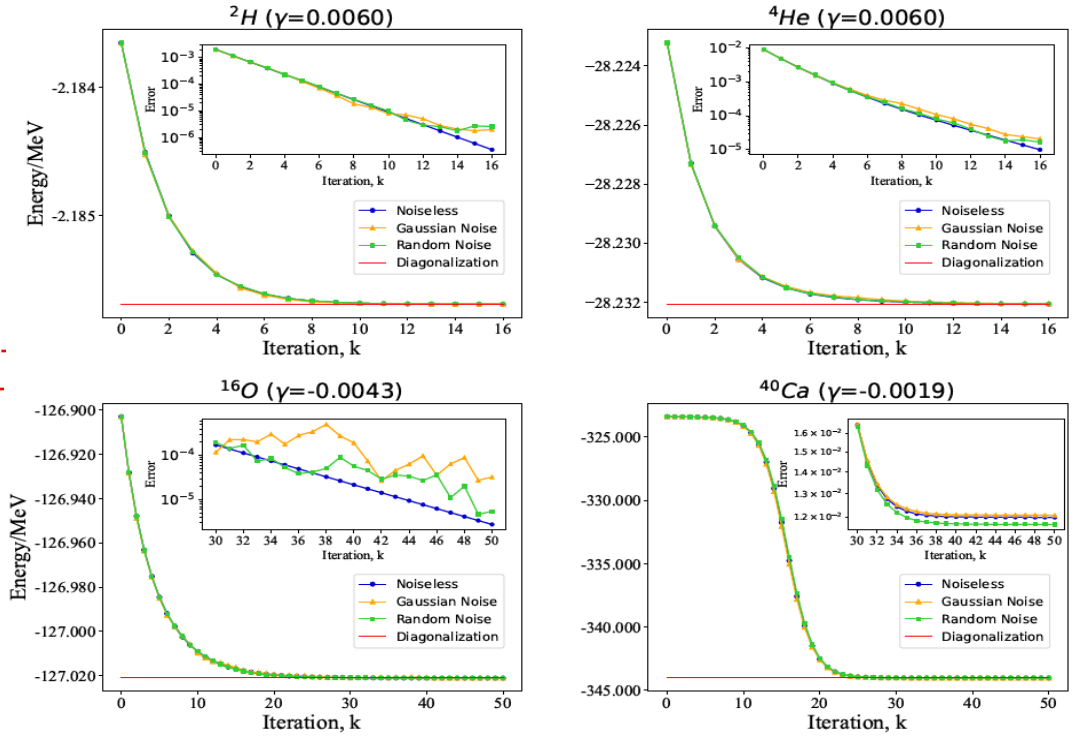
全量子本征求解器







实验演示氘核结合能计算



数值模拟基态能量

哈密顿量细节

初始哈密顿量

$$H=\sum_{i=1}^{Z}\frac{\vec{p}_i^2}{2M_p}+\sum_{j=Z+1}^{A}\frac{\vec{p}_j^2}{2M_n}+\sum_{k=1}^{A}V(\vec{r}_k)+$$
$$\sum_{1\le i_1<i_2\le Z}\frac{e^2}{4\pi\epsilon_0}\frac{1}{|\vec{r}_{i_1}-\vec{r}_{i_2}|}+\frac{1}{2}\sum_{k_1\neq k_2}v(k_1,k_2).$$

平均场参数

$$V(\vec{r})=U_c(r)-2\lambda\left(\frac{\hbar}{2Mc}\right)^2\cdot\frac{dU_c(r)}{dr}\vec{s}\cdot\vec{l},$$
$$U_c(r)=U_0(1\pm\kappa\frac{N-Z}{N+Z})/(1+\exp(\frac{r-r_0A^{1/3}}{a_0})).$$

二次量子化

$$H=\sum_{\alpha'}\sum_{\alpha}g_\pi(\alpha',\alpha)a^\dagger_{\pi,\alpha'}a_{\pi,\alpha}\otimes I_\nu+$$
$$\sum_{\alpha_1'\alpha_2'}\sum_{\alpha_1\alpha_2}h_\pi(\alpha_1',\alpha_2',\alpha_1,\alpha_2)a^\dagger_{\pi,\alpha_1'}a^\dagger_{\pi,\alpha_2'}a_{\pi,\alpha_2}a_{\pi,\alpha_1}\otimes I_\nu+$$
$$\sum_{\beta'}\sum_{\beta}g_\nu(\beta',\beta)I_\pi\otimes a^\dagger_{\nu,\beta'}a_{\nu,\beta}+$$
$$\sum_{\beta_1'\beta_2'}\sum_{\beta_1\beta_2}h_\nu(\beta_1',\beta_2',\beta_1,\beta_2)I_\pi\otimes a^\dagger_{\nu,\beta_1'}a^\dagger_{\nu,\beta_2'}a_{\nu,\beta_2}a_{\nu,\beta_1}+$$
$$\sum_{\alpha'\beta'}\sum_{\alpha\beta}h_{\pi\nu}(\alpha',\alpha,\beta',\beta)a^\dagger_{\pi,\alpha'}a_{\pi,\alpha}\otimes a^\dagger_{\nu,\beta'}a_{\nu,\beta}.$$

二次量子化参数

$$g_\pi(\alpha',\alpha)=\langle\psi_{\pi,\alpha'}|\frac{\vec{p}^2}{2M_p}+V(\vec{r})|\psi_{\pi,\alpha}\rangle,$$
$$g_\nu(\beta',\beta)=\langle\psi_{\nu,\beta'}|\frac{\vec{p}^2}{2M_n}+V(\vec{r})|\psi_{\nu,\beta}\rangle,$$
$$h_\pi(\alpha_1',\alpha_2',\alpha_1,\alpha_2)=\langle\psi_{\pi,\alpha_1'}|\langle\psi_{\pi,\alpha_2'}|\frac{1}{2}\Big\{\frac{e^2}{4\pi\epsilon_0}\frac{1}{|\vec{r}_{\pi_1}-\vec{r}_{\pi_2}|}+u(\pi_1,\pi_2)\Big\}|\psi_{\pi,\alpha_1}\rangle|\psi_{\pi,\alpha_2}\rangle,$$
$$h_\nu(\beta_1',\beta_2',\beta_1,\beta_2)=\langle\psi_{\nu,\beta_1'}|\langle\psi_{\nu,\beta_2'}|\frac{1}{2}v(\nu_1,\nu_2)|\psi_{\nu,\beta_1}\rangle|\psi_{\nu,\beta_2}\rangle,$$
$$h_{\pi\nu}(\alpha',\alpha,\beta',\beta)=\langle\psi_{\pi,\alpha'}|\langle\psi_{\nu,\beta'}|v(\pi,\nu)|\psi_{\pi,\alpha}\rangle|\psi_{\nu,\beta}\rangle.$$

哈密顿量映射

编码量子态

$$\overbrace{|0100...101\rangle|1001...110\rangle}^{\underbrace{N_\pi \ qubits}\quad N_\nu \ qubits}$$
$$Z \text{ protons } N \text{ neutrons}$$

JW变换

$$\begin{aligned} a_k^\dagger &= c_k|1\rangle_k\langle 0| = \frac{c_k}{2}(X_k - iY_k) \\ &= \frac{1}{2}Z_1 \otimes Z_2 \otimes ...Z_{k-1} \otimes (X_k - iY_k) \otimes I_{k+1} \otimes ... \\ a_k &= c_k|0\rangle_k\langle 1| = \frac{c_k}{2}(X_k + iY_k) \\ &= \frac{1}{2}Z_1 \otimes Z_2 \otimes ...Z_{k-1} \otimes (X_k + iY_k) \otimes I_{k+1} \otimes ... \end{aligned}$$

Qubit形式哈密顿量

$$H = \sum_{k=0}^{M-1} \alpha_k H_k^g$$

Jordan-Wigner(J-W)变换

- To simulate a system of fermions on a quantum computer, we must choose a representation of the ladder operators on the Hilbert space of the qubits.
- In other words, we must designate a set of qubit operators (matrices) which satisfy the canonical anticommutation relations. Qubit operators are written iterms of the Pauli matrices X ,Y and Z. Jordan-Wigner(J-W)Transformation maps fermionic operators to qubit operators by the following rules:

$$a_p \mapsto \frac{1}{2}(X_p + \mathrm{i}Y_p)Z_1 \cdots Z_{p-1}$$

$$= (|0\rangle\langle 1|)_p Z_1 \cdots Z_{p-1}$$

简单的例子

- $a_1^\dagger\, a_2 = 0.25X_1X_2\ + 0.25j\ X_1Y_2\ - 0.25j\ \ Y_1X_2 + 0.25\ Y_1Y_2$
- $a_1 a_2^\dagger\ = -0.25X_1X_2\ + 0.25j\ X_1Y_2\ - 0.25j\ \ Y_1X_2 - 0.25\ Y_1Y_2$

The molecular Hamiltonian of H_2 is

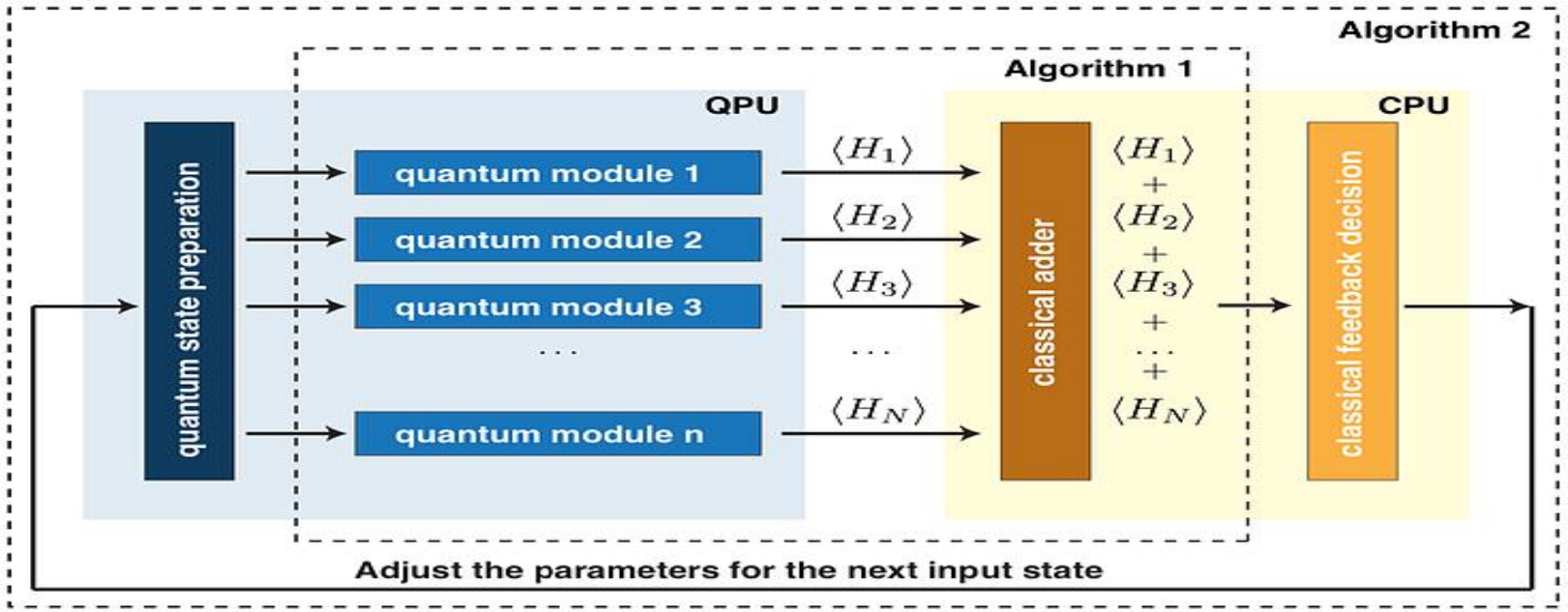
$$
\begin{aligned}
H =& c_0\mathcal{I} + c_1(\sigma_0^z + \sigma_1^z) + c_2(\sigma_2^z + \sigma_3^z) + c_3\sigma_3^z\sigma_2^z \\
& c_4\sigma_1^z\sigma_0^z + c_5(\sigma_2^z\sigma_0^z + \sigma_3^z\sigma_1^z) + c_6(\sigma_2^z\sigma_1^z + \sigma_3^z\sigma_0^z) \\
& + c_7(\sigma_3^x\sigma_2^y\sigma_1^y\sigma_0^z + \sigma_3^y\sigma_2^x\sigma_1^x\sigma_0^y) \\
& - c_7(\sigma_3^x\sigma_2^x\sigma_1^y\sigma_0^y + \sigma_3^y\sigma_2^y\sigma_1^x\sigma_0^x),
\end{aligned}
$$

c_i is coefficient related to $h_{ij}\ and\ \ h_{ijkl}$

变分量子算法 (VQE)

- The Variational-Quantum-Eigensolver (VQE) is a quantum-classical hybrid algorithm that can be used to find eigenvalues of a matrix H. In this hybrid algorithm a quantum subroutine is run inside of a classical optimization loop.
- The quantum subroutine has two fundamental steps:

1. Prepare the quantum state $|\Psi(\text{vec}(\theta))\rangle|\Psi(\text{vec}(\theta))\rangle$, often called the ansatz.
2. Measure the expectation value $\langle\Psi(\text{vec}(\theta))|H|\Psi(\text{vec}(\theta))\rangle$.



FQE的基本原理

Gradient descent iterations: $X^{t+1} = X^t - \gamma \nabla f(X^t)$

Objective function (the expectation value of H): $f(X) = X^T H X$

Assume X is a real vector $\nabla f(X) = 2HX$

Quantum version: $|X^{t+1}\rangle = |X^t\rangle - \gamma H |X^t\rangle$

LCU form $|\mathbf{X}^{(t+1)}\rangle = \mathbf{H}^g |\mathbf{X}^{(t)}\rangle = \sum_{i=1}^{M} \beta_i \mathbf{H}_i^g |\mathbf{X}^{(t)}\rangle$

应用于原子核模拟的量子线路

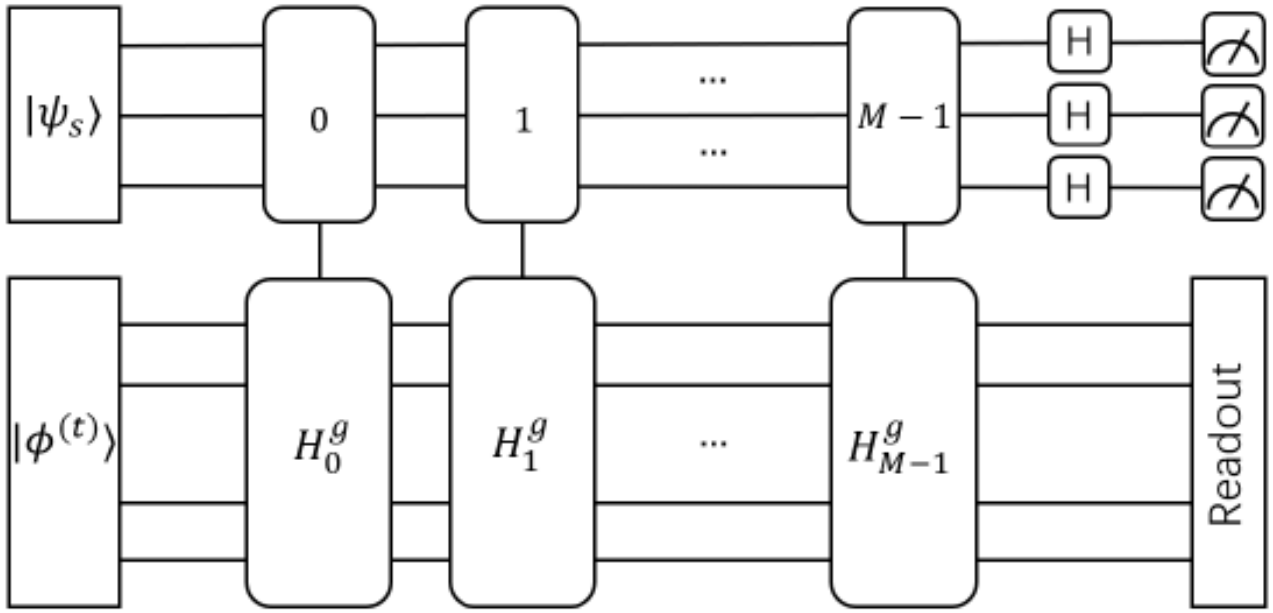


Figure 1. Quantum circuit of QCSH. Here, H is the Hadamard gate, and $H_k^g (0 \leq k \leq M-1)$ are the unitary operators composed of Pauli operators.

$$Sucessful\ probability$$

$$\boldsymbol{P_s} = \| \boldsymbol{H^g}|\boldsymbol{\phi(t)}\rangle \|^2 / \boldsymbol{MC^2}.$$

演化过程

- ***Create Superposition.***

The register is a composite system which contains a work system $|x\rangle$ *and an m-qubit ancillary register* $|\psi_s\rangle$.

$$|\Phi\rangle = |\psi_s\rangle|\phi\rangle = \frac{1}{C}\sum_{i=0}^{M-1}\beta_i|k\rangle|\phi\rangle,$$

where C is a normalization constant

- ***Entanglement.***

A series of ancillary system controlled operations $\sum_{k=0}^{M-1}|k\rangle\langle k|\otimes H_i^g$ *are implemented on the work qubits.*

$$|\Phi\rangle \rightarrow \frac{1}{C}\sum_{k=0}^{M-1}\beta_i|k\rangle H_i^g|\phi\rangle$$

- ***Combination.***

we perform m Hadamard gates on ancillary register to combine all the wave functions from the M different subspaces. We merely focus on the component in a subspace where the ancillary system is in state 0.

$$|\Phi_0\rangle \rightarrow \frac{1}{C\sqrt{2^m}}\sum_{k=0}^{M-1}|0\rangle\beta_k H_k^g|\phi\rangle$$

- ***Measurement.***

We measure the ancillary register. If we obtain 0, our algorithm succeeds and we obtain the work system in state.

$$\left|\Phi_f\right\rangle = H^g|\phi\rangle.$$

$$\boldsymbol{P_s} = \| \boldsymbol{H^g|\phi(t)\rangle} \|^2 / \boldsymbol{MC^2}. \quad Sucessful\ probability$$

误差估计与复杂度

However, the measurements of expectation values during the iteration procedure will destroy the final state of the work system, making our method difficult to realize. So, determine the iteration depth k in advance is essential. After k times iterations, the approximation error is limited to (ignoring constants)

$$\epsilon \leq O\left(\left(\frac{1-\gamma\lambda_2}{1-\gamma\lambda_1}\right)^k N\right) \qquad k = O(log\frac{N}{\epsilon})$$

λ1 and λ2 are the two eigenvalues of largest absolute value of Hamiltonian H

$$P_s < \frac{1}{M}\frac{(\lambda_{d-1}-\lambda_0)^2}{(\frac{1}{2\gamma}-\alpha_0)^2+\sum_{k=1}^{M-1}|\alpha_k|^2},$$
$$P_s > \frac{|c_0^2|}{4M}\frac{(\lambda_{d-1}-\lambda_0)^2}{(\frac{1}{2\gamma}-\alpha_0)^2+\sum_{k=1}^{M-1}|\alpha_k|^2}.$$

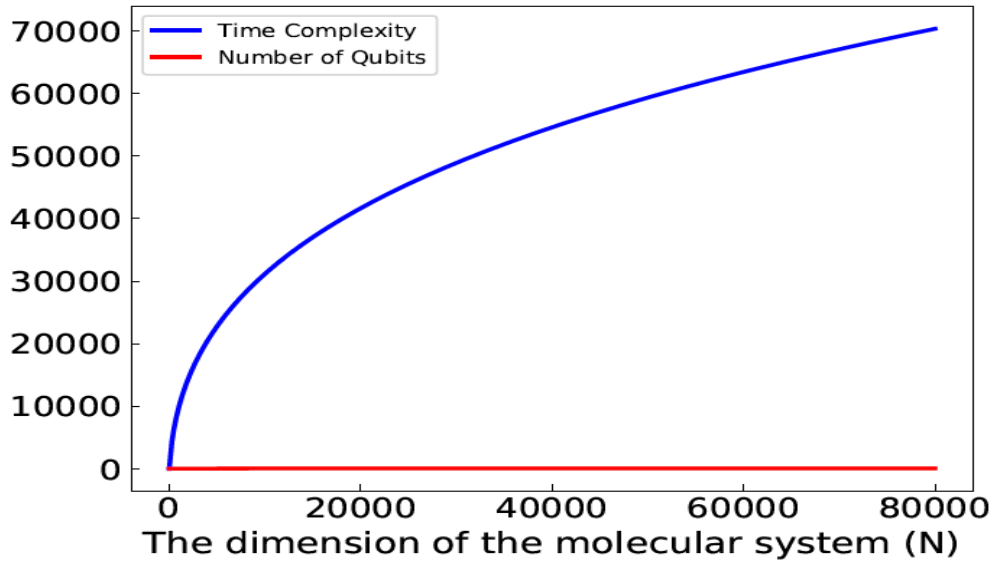


Figure 5: The number of qubits and basic gates required to simulate molecules with the dimension increasing.

In the circumstances that the wavefunction is expressed by O(N) Gaussian orbitals, Fermion Hamiltonians contain $O(N^4)$ second-quantized terms, consequently qubit Hamiltonians with $O(N^4)$ Pauli terms. The qubit resource and gate complexity can be reduced to O(N) and $O(N^4)$ respectively.

微扰理论

$$H = \sum_{i,\alpha} h^i_\alpha \sigma^i_\alpha + \sum_{i,j,\alpha,\beta} h^{ij}_{\alpha\beta} \sigma^i_\alpha \sigma^j_\beta + \dots$$

$$H = H_0 + H'$$

σ_Z and identity matrices　　others

$$\mathbf{H}|\psi_n\rangle = \left(\mathbf{H}_0 + \mathbf{H}'\right)|\psi_n\rangle = E_n|\psi_n\rangle,$$

zero. First order corrections can be simplified as

$$E_n = E_n^{(0)} \tag{17}$$

$$|\psi_n\rangle = |n\rangle - \sum_{m\neq n} \frac{H'_{mn}}{E_m^{(0)} - E_n^{(0)}}|m\rangle \tag{18}$$

and second-order corrections can be rewritten as

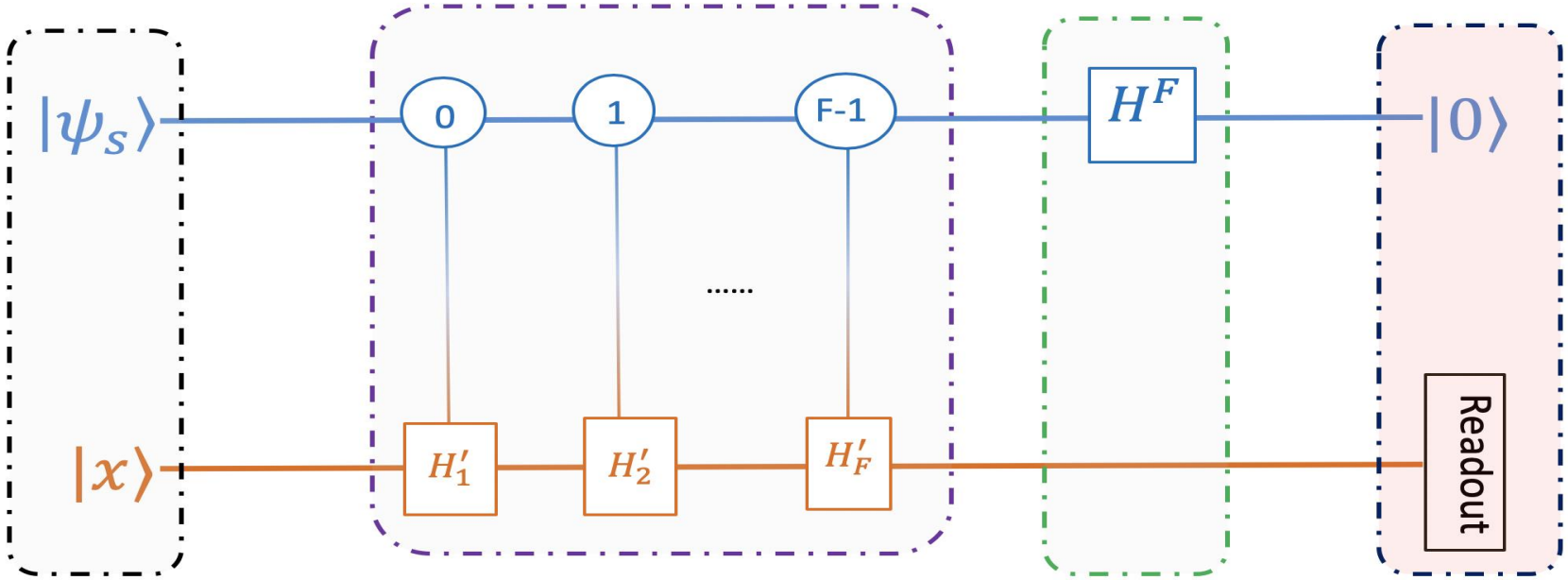
$$E_n = E_n^{(0)} + \sum_{m\neq n} \frac{|H'_{mn}|^2}{E_m^{(0)} - E_n^{(0)}} \tag{19}$$

$$|\psi_n\rangle = |n\rangle - \sum_{m\neq n} \frac{H'_{mn}}{E_m^{(0)} - E_n^{(0)}}|m\rangle - \frac{1}{2}\sum_{m\neq n} \frac{|H'_{mn}|^2}{(E_m^{(0)} - E_n^{(0)})^2}|n\rangle + \sum_{m\neq n}\left[\sum_{k\neq n} \frac{H'_{mn}H'_{kn}}{(E_m^{(0)} - E_n^{(0)})(E_k^{(0)} - E_n^{(0)})}\right]|m\rangle \tag{20}$$

一次迭代得到H'_{mn}

The matrix elements in the first and second-order approximations can be obtained by one time iteration of quantum circuit in Fig.(1). Here, we let $\mathbf{H}'$ be equal to $\mathbf{H}^g$. Explicitly, the first order approximation only involves $\mathbf{H}'_{mn}$, a series of transition probabilities of the state after $\mathbf{H}'$ implemented on state $|n\rangle$, and they can be obtained by performing the quantum circuit of Fig.(1) directly. For the second order approximation, matrix elements such as value $|\mathbf{H}'_{mn}|^2$ and $\mathbf{H}'_{mn}\mathbf{H}'_{kn}$, can be calculated by $\mathbf{H}'_{mn}$. Then, the approximate ground energy and ground state can be obtained by only one time iteration.

$$H' = H'_1 + H'_2 + \ldots + H'_F$$



寻找最稳定结构

- If we want to calculate the interatomic distance corresponding to the most stable structure, the variation of interatomic distances is necessary.
- The lowest energy in potential-energy surfaces corresponds to the most stable structure of the molecules.
- four examples are given to illustrate the performance of the perturbation theory.

性能

- As shown in the picture, the ground-state energy of each molecule calculated under the second-order approximation is already quite close to its exact value.

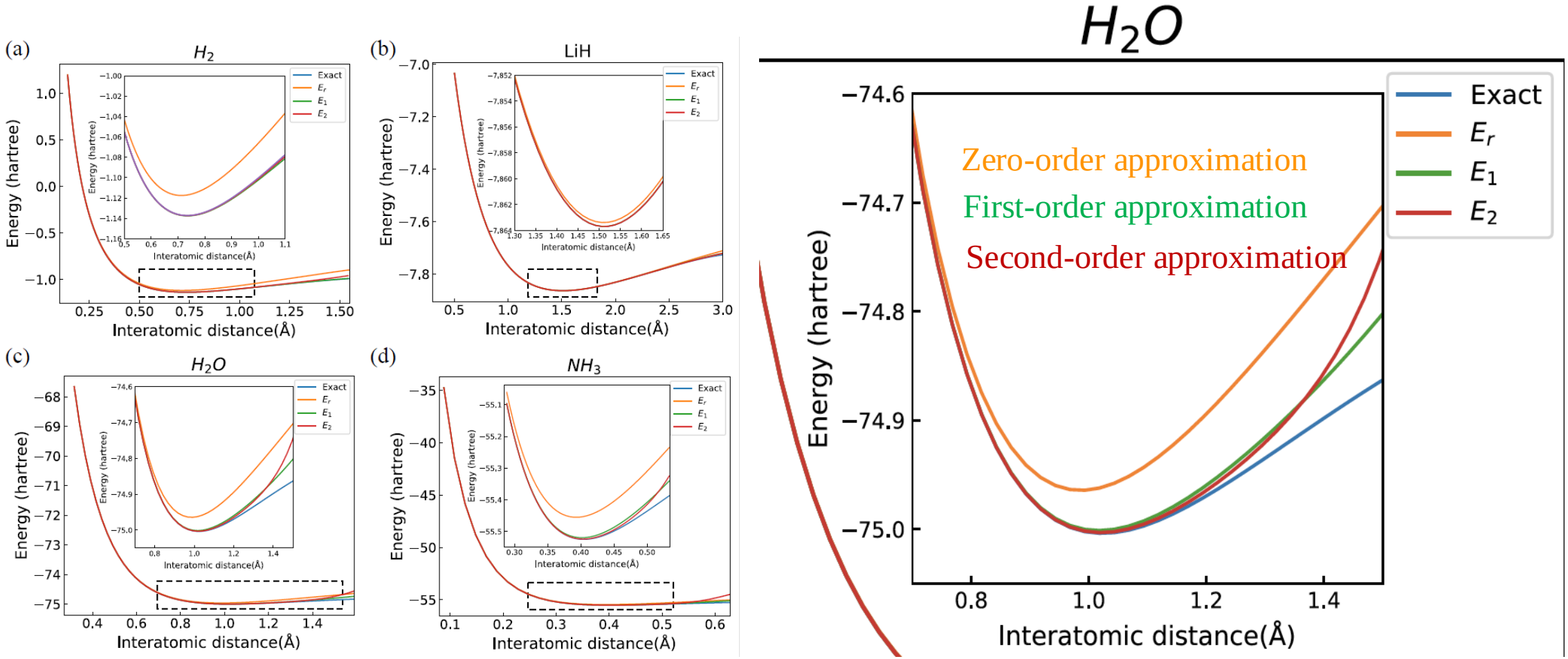
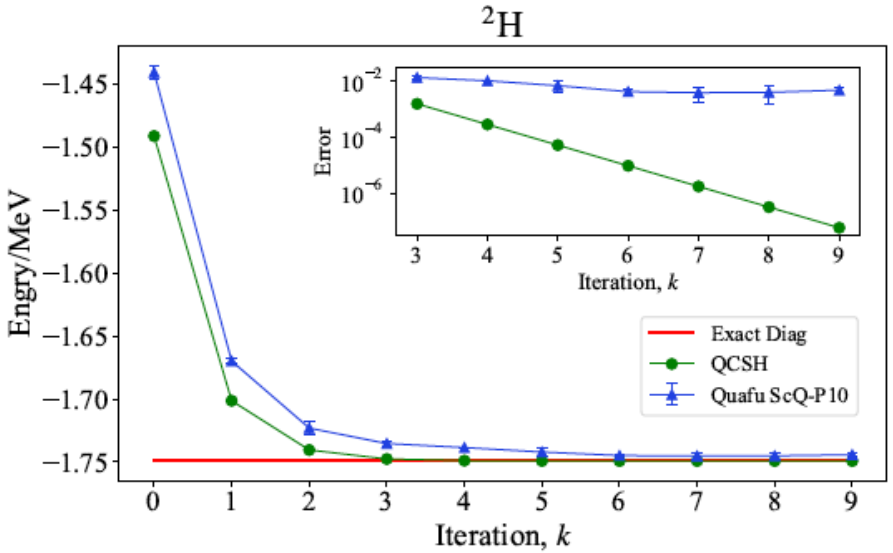


Figure : Theory result (blue lines), zero-order (orange lines), first-order (green lines) and second-order (red lines) energy plots of outcomes from numerical simulations, for several interatomic distances for H_2(a), LiH(b) , H_2O(c), and NH_3(d).

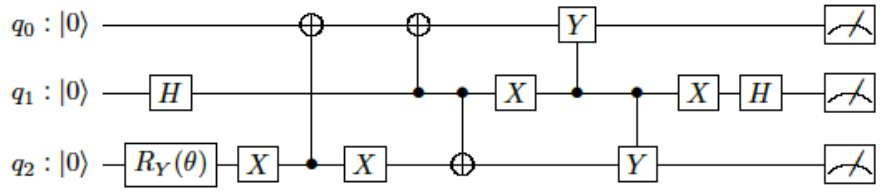
算法的实验实现

利用FQE在超导量子计算机上计算氘核的结合能

$$H_2 = 5.906709I + 0.218291Z_0 - 6.125Z_1 \\ -2.143304\,(X_0X_1 + Y_0Y_1)$$



(b)



目前关注

- 粒子流事例重建
- 热传导方程相关的输运过程
- 强关联系统量子模拟.

重建基本粒子的事例--组合优化问题

研究物理过程：通过区分对撞事件，我们可以研究粒子之间的相互作用、衰变过程、能量转移等物理过程。这有助于验证和完善我们对基本相互作用的理论模型，同时也为理解宇宙演化、宏观现象提供重要线索。对于本题目粒子流配对问题，研究A 集合里的粒子流与B 集合里的粒子流之间的匹配，通过与强子化前基本粒子能量的匹配，重建基本粒子的事例。

热传导量子算法(QHC)

研究背景

热传导的数学方程是一个典型的偏微分方程计算复杂度高

解决方案

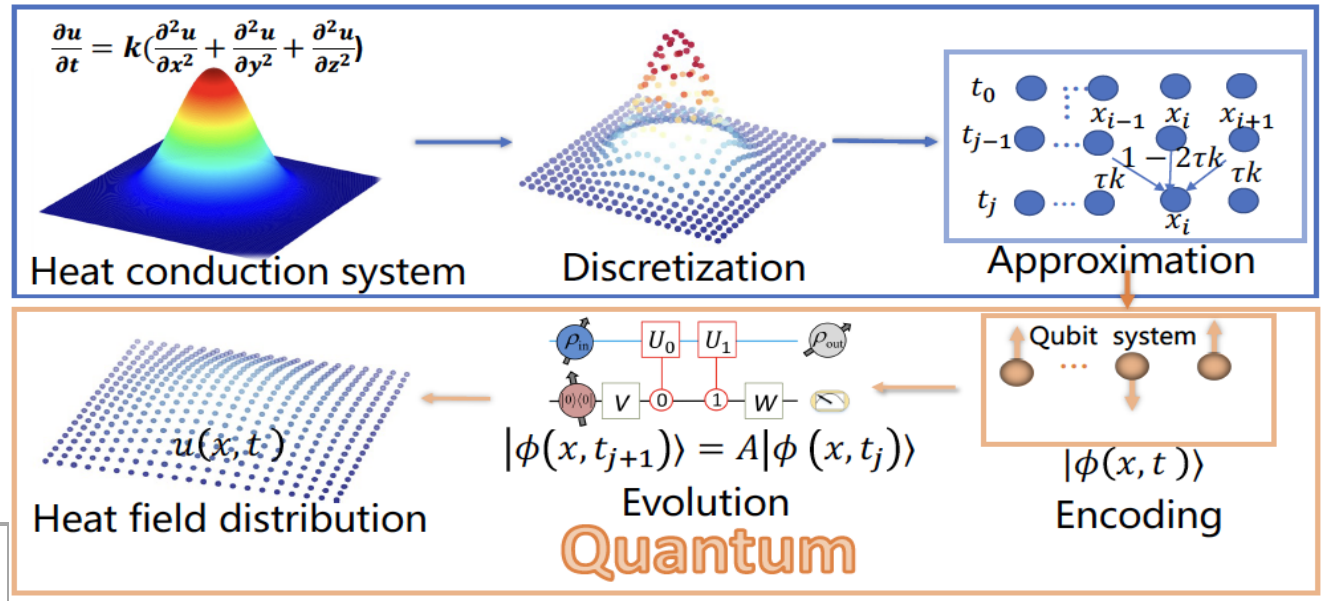
提出了一种优于经典算法的热传导（QHC）量子算法，量子电路的复杂性与离散化网格点的数量成多项式对数关系，并在核磁平台上进行了实验验证。

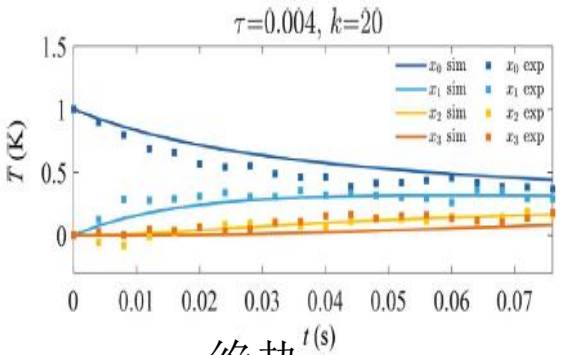
特点优势

1. 用一个带有辅助量子比特的对称系统来表示原始热传导系统，量子电路的复杂性就离散化网格点的数量而言是对数
2. 与基于通过 Harrow-Hassidim-Lloyd (HHL) 算法求解线性方程的现有算法相比，该方法直接演化线性过程而无需相位估计

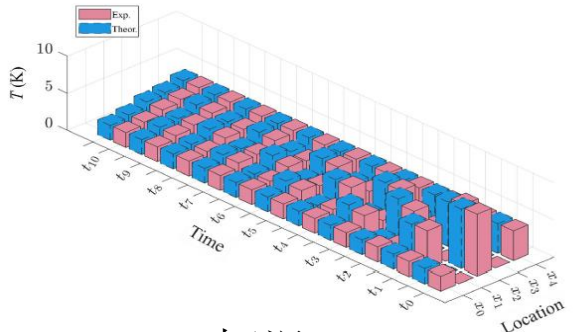
应用领域

自然地应用于可以简化为热方程的物理过程、如航天器和稀释制冷机等





绝热



恒温

Wei S J, Wei C, Lv P, et al. A quantum algorithm for heat conduction with symmetrization[J]. Science Bulletin, 2023, 68(5): 494-502.[第一作者]

热传导量子算法(QHC)

热传导方程离散化并近似

$$\frac{\mathrm{d}u}{\mathrm{d}t}=k\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)$$

$$R=[0,\ M]^3\times[0,\ T]$$

$$\begin{aligned}\frac{\mathrm{d}u}{\mathrm{d}t}&=k\frac{d^2u}{\mathrm{d}x^2}\\&=k\frac{u(x+h)+u(x-h)-2u(x)}{h^2}\\&\quad+\frac{h^2k}{24}\left(\frac{\mathrm{d}^4u}{\mathrm{d}x^4}(\xi')+\frac{\mathrm{d}^4u}{\mathrm{d}x^4}(\xi'')\right)\end{aligned}$$

一维，自变量变换 $u(x_i,t)=v(x_i,t)+mi.$

$$\begin{aligned}v(x_i,\ t_{j+1})&=v(x_i,\ t_j)+\tau\frac{\mathrm{d}v(x_i,\ t)}{\mathrm{d}t}\\&=\tau k\,v((x_{i-1},\ t_j)+v(x_{i+1},\ t_j))\\&\quad+\left(\frac{1-2\tau k}{\tau k}\right)v(x_i,\ t_j).\end{aligned}$$

对应演化过程 $v(\mathbf{x},\ t_{j+1})=\mathbf{A}v(\mathbf{x},\ t_j)$

$$\begin{bmatrix}1&0&0&&0\\\tau k&1-2\tau k&\tau k&\dots&0\\0&\tau k&1-2\tau k&&0\\&\dots&&&\\0&0&0&\dots&1\end{bmatrix}.$$

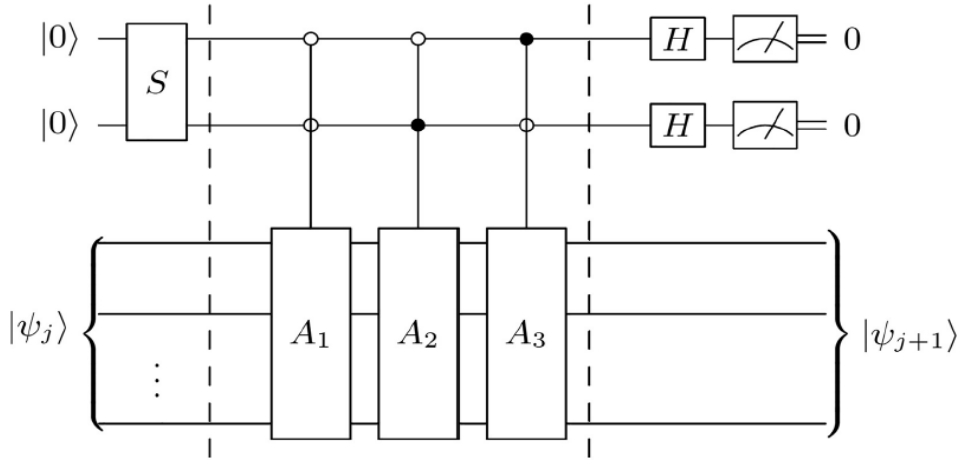
离散点对称化 $v\left(\bar{x}_i,\ t_j\right)=2v(x_M,\ t_j)-v(x_{M-i},\ t_j)$

$$\hat{\mathbf{A}}=\begin{bmatrix}1-2\tau k&\tau k&0&&0&\tau k\\\tau k&1-2\tau k&\tau k&\dots&0&0\\0&\tau k&1-2\tau k&&\tau k&0\\&\dots&&&&\\\tau k&0&0&\dots&\tau k&1-2\tau k\end{bmatrix}.$$

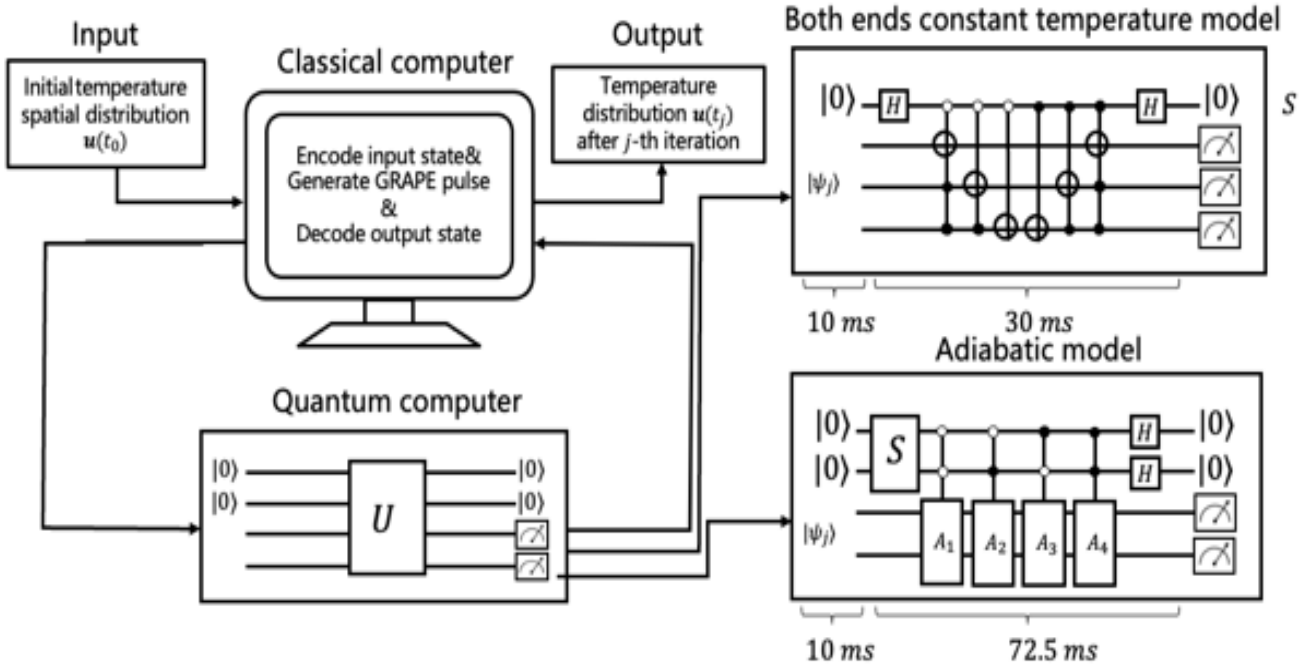
热传导量子算法(QHC)

LCU实现

$$(1-2\tau k)\mathbf{A_1}+\tau k\mathbf{A_2}+\tau k\mathbf{A_3}$$



实验演示方案



$$S=\frac{1}{\sqrt{(1-\tau k)^2+3(\tau k)^2}}\begin{pmatrix}1-\tau k & \tau k & \tau k & -\tau k\\ \tau k & \tau k & \tau k-1 & \tau k\\ \tau k & \tau k-1 & -\tau k & -\tau k\\ -\tau k & \tau k & -\tau k & \tau k-1\end{pmatrix}$$

$$A_1=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}\quad A_2=\begin{pmatrix}0&0&0&1\\1&0&0&0\\0&1&0&0\\0&0&1&0\end{pmatrix}$$

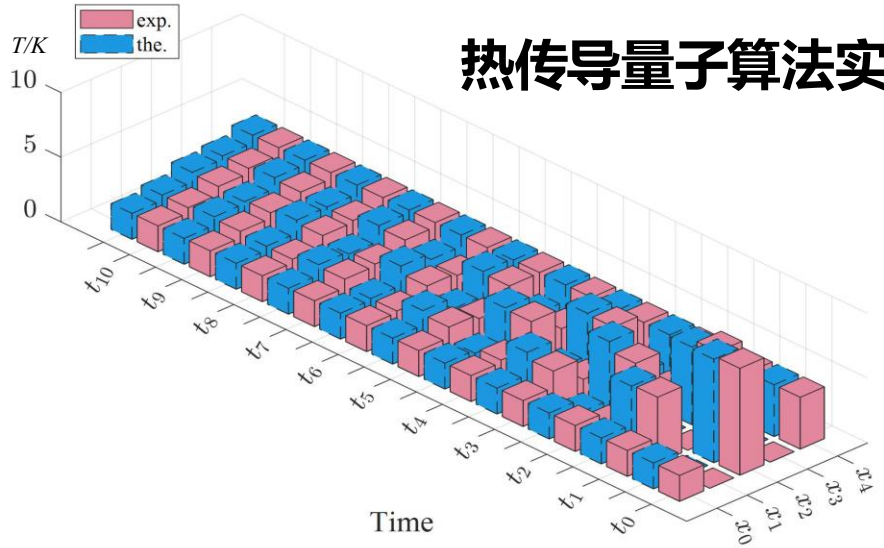
$$A_3=\begin{pmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0\end{pmatrix}\quad A_4=\begin{pmatrix}0&0&0&1\\0&1&0&0\\0&0&1&0\\1&0&0&0\end{pmatrix}$$

4比特，恒温，绝热

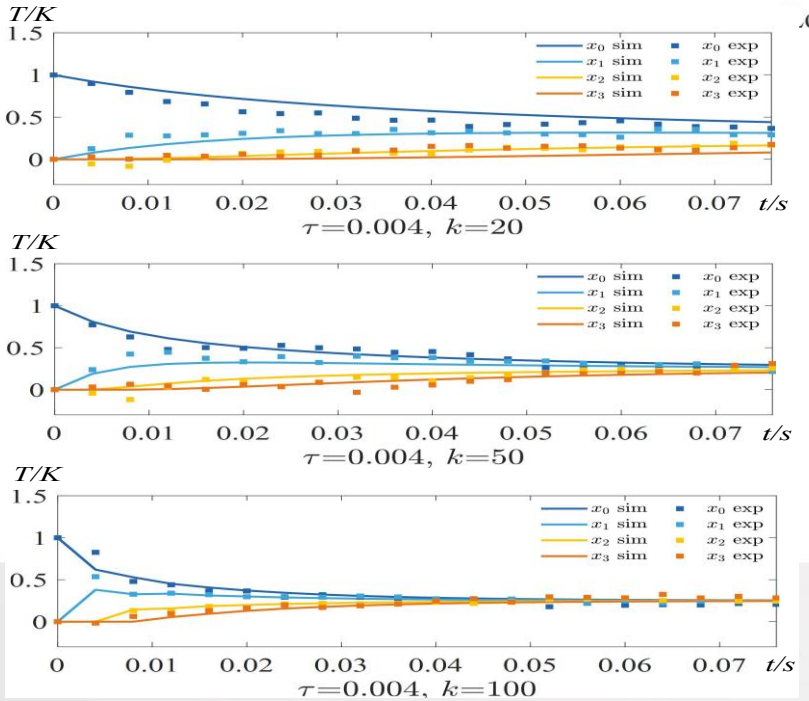
热传导量子算法(QHC)

热传导量子算法实验展示

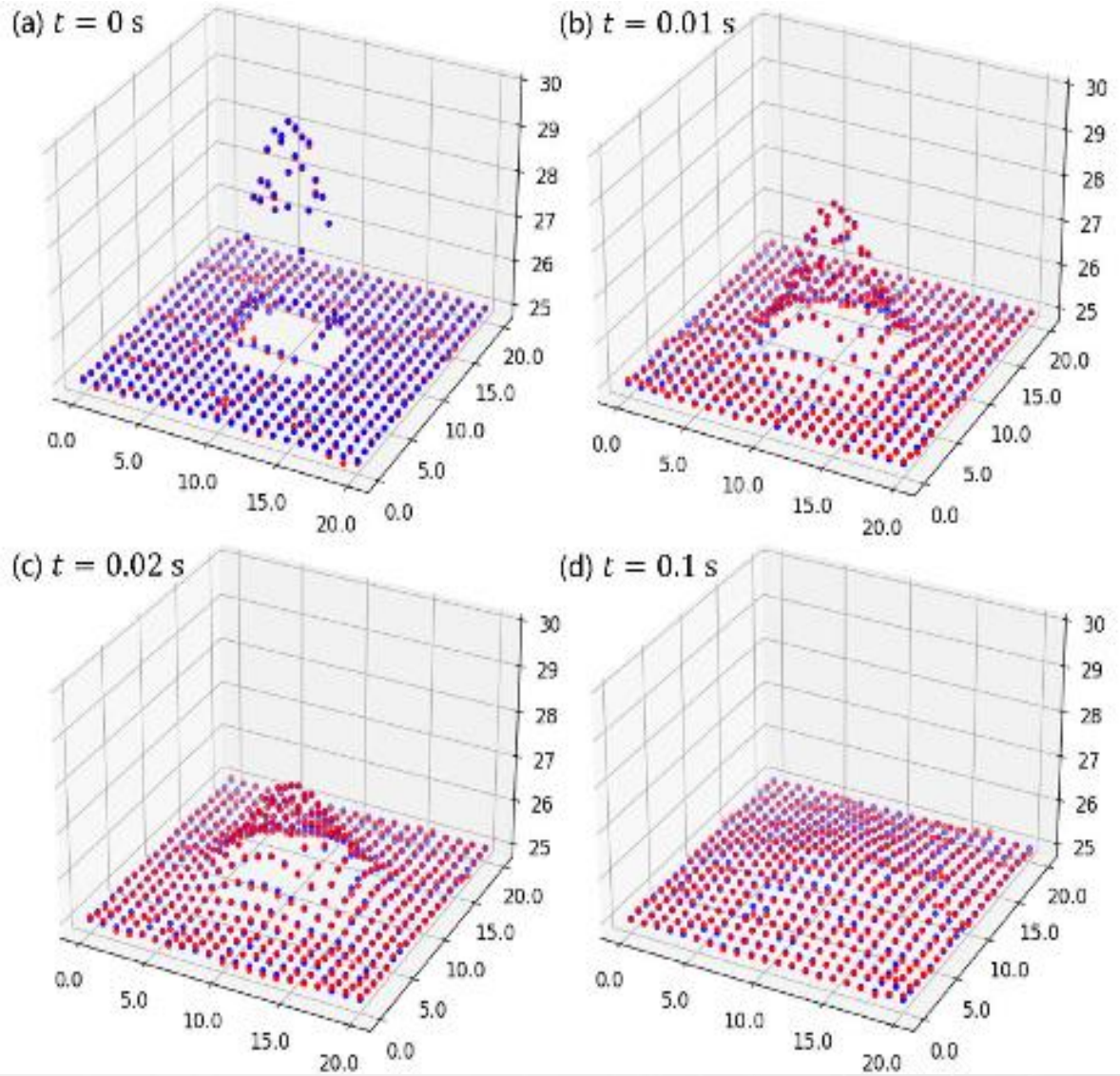
两端恒温



两端绝热



蓝色为经典模拟，红色为量子算法模拟



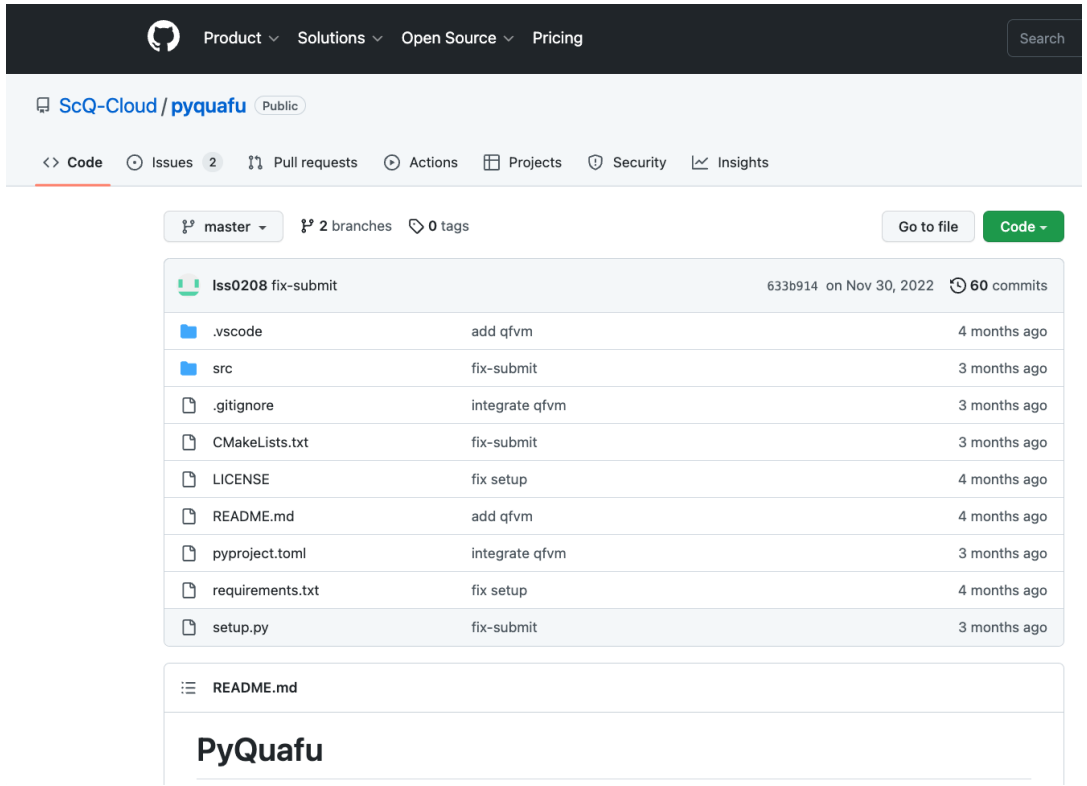
超导量子云平台——Quafu

Quafu云平台网页

quafu.baqis.ac.cn

python工具包 PyQuafu

https://github.com/ScQ-Cloud/pyquafu

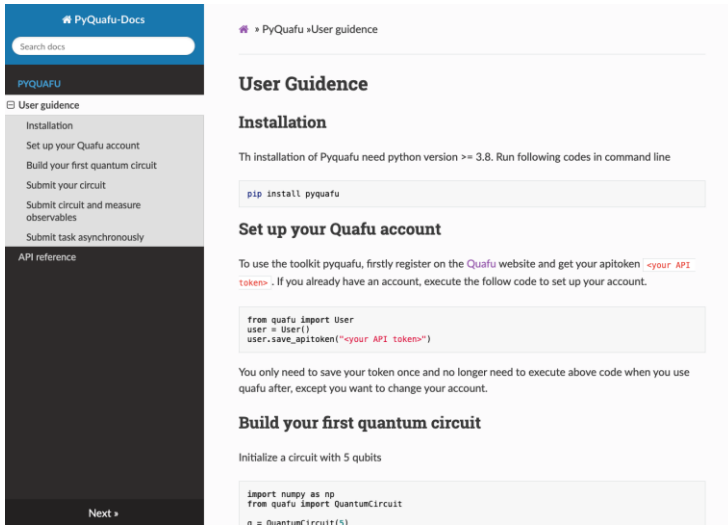


安装方式：

1.通过pip安装

pip install pyquafu

2.GitHub源安装



文档地址：https://scq-cloud.github.io/

谢谢大家！